Examiner Isaac:

Per our conversation of March 25, 2008, please find a complete listing of the claims which incorporate the proposed amendments of March 20, 2008.

1. (Currently Amended) A method in a data processing system, for code reusability and maintainability, the method comprising:

providing a utility class in a server, wherein the utility class defines a utility method, the utility method being written in an object oriented programming language;

receiving a markup language request at the server for an entity from a client, the markup language request including a response object name, wherein the response object name is a string representing a desired element name for a markup language response object;

responsive to receiving the markup language request at the server for the entity from the client, generating a method call for the utility method, wherein the method call identifies the entity and the response object name;

generating [[a]] the markup language response object and assigning the response object name to the markup language response object; and

returning the markup language response object named as the response object name to the client.

2. (Canceled)

3. (Previously Presented) The method of claim 1, wherein the markup language request is an extensible markup language request.

4. (Original) The method of claim 3, wherein the extensible markup language request is one of a list request and a get request.

5. (Previously Presented) The method of claim 1, further comprising:

retrieving, by the utility method, at least one data item for the method call and the entity, wherein the markup language response object includes the at least one data item.

6. (Original) The method of claim 5, wherein the step of retrieving at least one data item includes retrieving the at least one data item from a database.

7. (Original) The method of claim 6, wherein the at least one data item is retrieved from the database through a structured query language interface.

8. (Previously Presented) The method of claim 5, wherein the markup language request includes a list of attributes.

9. (Original) The method of claim 8, wherein the at least one data item includes a set of attributes for the entity, wherein the set of attributes corresponds to the list of attributes.

10. (Original) The method of claim 9, wherein the list of attributes is an empty string.

11. (Original) The method of claim 10, wherein the set of attributes includes all attributes for the entity.

12. (Previously Presented) The method of claim 1, wherein the markup language response object is an extensible markup language document.

13. (Previously Presented) A data processing system[[,]] for code reusability and maintainability, the data processing system comprising:

a bus;

a communications unit connected to the bus;

a storage device connected to the bus, wherein the storage device includes computer usable program code; and

a processor unit connected to the bus, wherein the processor unit executes the computer usable program code to:

provide a utility class, wherein the utility class defines a utility method, the utility method being written in an object oriented programming language;

receive a markup language request at the server for an entity from a client, the markup language request including a response object name, wherein the response object name is a string representing a desired element name for a markup language response object;

responsive to receiving the markup language request for attributes for the entity from the client, generate a method call for the utility method, wherein the method call identifies the entity and the response object name;

generate [[a]] the markup language response object and ~~assigns~~ assign the response object name to the markup language response object; and

return the markup language response object named as the response object name to the client.

14. (Previously Presented) The data processing system of claim 13, wherein the client includes an extensible markup language interface and wherein the markup language request is an extensible markup language request.

15. (Previously Presented) The data processing system of claim 13, wherein the utility method retrieves at least one data item for the method call and the entity and wherein the markup language response object include the at least one data item.

16. (Previously Presented) The data processing system of claim 15, wherein the utility method retrieves the at least one data item from a database.

17. (Previously Presented) The data processing system of claim 15, wherein the markup language request includes a list of attributes.

18. (Currently Amended) The data processing system of claim 17, wherein the at least one data item includes a set of attributes for the entity, wherein the set of attributes corresponds to the list of attributes.

19. (Previously Presented) The data processing system of claim 18, wherein the list of attributes is an empty string.

20. (Previously Presented) The data processing system of claim 19, wherein the set of attributes includes all attributes for the entity.

21. (Previously Presented) The data processing system of claim 13, wherein the markup language response object is an extensible markup language document.

22. (Currently Amended) A computer program product, in a computer readable physical storage medium, for code reusability and maintainability, the computer program product comprising:

instructions, in a utility class, for defining a utility method, the utility method being written in an object oriented programming language;

instructions for receiving a markup language request at the server for an entity from a client, the markup language request including a response object name, wherein the response object name is a string representing a desired element name for a markup language response object;

instructions, responsive to receiving the markup language request at the server for attributes for the entity from the client, for generating a method call for the utility method, wherein the method call identifies the entity and the response object name;

instructions for generating [[a]] the markup language response object and assigning the response object name to the markup language response object; and

instructions for returning the markup language response object named as the response object name to the client.

23. (Previously Presented) The method of claim 1, wherein the server is located at a first computer system, wherein the client is located at a second computer system, and wherein the first computer system is separate from the second computer system.